AMENDMENTS TO THE CLAIMS

Kindly replace the claims as follows.

1. (currently amended) A method, comprising:

during a profiled interval of an execution of a program on a computer, recording profile information describing the execution, ~~without the program having been compiled for profiled execution,~~ the program being coded in an instruction set in which instructions are not all of the same length, and in which the length ~~an interpretation~~ of an instruction depends on a processor mode not expressed in the binary representation of the instruction,

the recorded profile information containing information sufficient to permit, without reference to the binary representation of the program, reliable inference of the address of the last byte of a multi-byte control transfer instruction ~~describing at least all events occurring during the profiled execution interval of the two classes~~:

~~a divergence of execution from sequential execution;~~

~~a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change taken together with a processor mode before the mode change instruction;~~

~~the profile information further identifying each distinct physical page of instruction text executed during the execution interval~~.

2. (original) The method of claim 1:

and further comprising, without software intervention, recording for later analysis a profile entry noting the source and destination of a control flow event in which control flow of the program execution diverges from sequential execution.

3. (original) The method of claim 1, wherein the program is executed on a computer having:

an instruction pipeline configured to execute instructions from a memory of the computer; and

profile circuitry configured to detect the occurrence of profileable events occurring in the instruction pipeline, and to direct the instruction pipeline to record profile information describing

the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

4. (original) The method of claim 3, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

5. (previously presented) The method of claim 3, the recorded profile information being efficiently tailored for storage in the memory of the computer to identify all bytes of object code executed during the profiled execution interval, without reference to the binary code of the program.

6. (original) The method of claim 3, wherein the recorded profile information describes a sequence of program events during the profiled interval of execution, the sequence including every event that matches time-independent criteria of profileable events to be profiled.

7. (original) The method of claim 3, further comprising:

when an instruction fetch of an instruction causes a miss in a translation look aside buffer (TLB), the fetch of the instruction triggering a profileable event, servicing the TLB miss and reflecting the corrected state of the TLB in the profile information recorded for the profileable instruction.

8. (original) The method of claim 1, further comprising the steps of:

executing the program on a first CPU of a multiprocessor computer;

on a second CPU of the multiprocessor, while the execution and profiling of the program continues, analyzing the collected profile data;

controlling the execution of the program on the first CPU based at least in part on the analysis of the collected profile data.

9. (original) The method of claim 1, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

10. (original) The method of claim 9, wherein the recorded profile information describes a sequence of program events during the profiled interval of execution, the sequence including every event that matches time-independent criteria of profileable events to be profiled.

11. (original) The method of claim 1:

wherein the program has been compiled without special consideration for execution profiling;

wherein the program execution induces occurrence of events that match time-independent criteria of profileable events to be profiled, the criteria including at least some physical memory references referenced by the program;

and further comprising, during a profile-quiescent interval of execution, recording no profile information in response to the occurrence of profileable events; and

commencing the profiled execution interval after a triggering event is detected, the triggering event being one of a predefined class of triggering events, the recorded profile information describing every event that matches the profileable event selection criteria induced after the triggering event, the recording continuing until a predetermined stop condition is reached.

12. (previously presented) The method of claim 11, wherein the recorded profile information is efficiently tailored for storage in the memory of the computer to identify all bytes of object code executed during the profiled interval, without reference to the binary code of the program.

13. (original) The method of claim 11, further comprising:

dividing the criteria for profileable events into initiating events and non-initiating events;

after the triggering event is detected, ignoring non-initiating profileable events; and

when an initiating event is detected, commencing recording the profile entries in the memory, describing every initiating and non-initiating event matching the profileable criteria during an interval following the triggering event.

14. (original) The method of claim 1, further comprising:

commencing the profiled execution interval at the expiration of a timer, the recorded profile describing a sequence of events including every event that matches time-independent selection criteria of events to be profiled, the recording continuing until a predetermined stop condition is reached.

15. (original) The method of claim 1, wherein the recorded physical memory references include addresses of binary instructions referenced by an instruction pointer, and at least one of the recorded instruction references records the event of a sequential execution flow across a page boundary in the address space.

16. (original) The method of claim 15, wherein at least one of the recorded instruction references records the event of a page boundary of the address space occurring within a single instruction.

17. (original) The method of claim 15, wherein the recorded profile information includes a record denoting sequential flow across a page boundary lying between two instructions that are sequentially adjacent in the logical address space..

18. (original) The method of claim 1, further comprising:

recording profile information recording a data-dependent change to a full/empty mask for registers of the computer.

19. (currently amended) A computer, comprising:

an instruction pipeline configured to execute instructions of the computer, the instructions being instructions of ~~program being coded in~~ an instruction set in which instructions are not all of the same length, and in which the length ~~an interpretation~~ of an instruction depends on a processor mode not expressed in the binary representation of the instruction;

profile circuitry configured to detect ~~, without compiler assistance for execution profiling,~~ the occurrence of profileable events occurring in the instruction pipeline, and to direct recording of profile information describing the detected profileable events, the recorded profile information containing information sufficient to permit, without reference to a binary representation of a profiled program, reliable inference of the address of the last byte of a multi-byte control transfer instruction ~~describing at least all events occurring during a profiled execution interval of the two classes:~~

~~a divergence of execution from sequential execution~~;

~~a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change taken together with a processor mode before the mode change instruction;~~

~~the profile information further identifying each distinct physical page of instruction text executed during the execution interval~~.

20. (original) The computer of claim 19, wherein:

the profile circuitry is configured to detect the occurrence of profileable events occurring in the instruction pipeline, and to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

21. (original) The computer of claim 20, wherein:

the profile circuitry is configured to record profile information into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

22. (original) The computer of claim 19:

wherein the instruction pipeline and profile circuitry are components of a first CPU of a multiprocessor;

and further comprising a second CPU of the multiprocessor, configured to analyze the recorded profile information while the execution and profiling of the program continues on the first CPU, and to at least partially control the operation of the first CPU based at least in part on the analysis of the collected profile data.

23. (original) The computer of claim 19, wherein:

the profile circuitry is configured to record profile information into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

24. (previously presented) The computer of claim 19, further comprising:

profile control bits implemented in the computer hardware, values of the profile control bits controlling a resolution of the operation of the profile circuitry;

a binary translator configured to translate programs coded in the instruction set in which an interpretation of an instruction depends on a processor mode not expressed in the binary representation of the instruction, being a first instruction set architecture, into instructions of a second instruction set architecture;

a profile analyzer configured to analyze the recorded profile information, and to set the profile control bits to values to improve the operation of the binary translator.

25. (previously presented) The computer of claim 19, wherein the profile circuitry is configured to record profile information efficiently tailored for storage in the memory of the computer to identify all bytes of object code executed during the profiled interval, without reference to the binary code of the program.

26. (original) The computer of claim 19, wherein:

the profile circuitry is designed to record profile information that records a sequence of events of the program during the profiled execution interval, the sequence including every event that matches time-independent criteria of profileable events to be profiled.

27. (original) The computer of claim 19, wherein:

the profile circuitry is designed based on a division of the profileable events into initiating events and non-initiating events;

triggering circuitry of the profile circuitry is designed to recognize a triggering event, and then to ignore non-initiating events in favor of one of the initiating events; and

the profile circuitry is designed to commence to record the profile entries in the memory, describing initiating and non-initiating events, after the initiating event is detected.

28. (original) The computer of claim 19, wherein the profile circuitry includes a timer interval value, specifying a frequency at which the profile circuitry is to monitor the instruction pipeline for profileable events.

29. (original) The computer of claim 19:

wherein the instruction pipeline is configured to execute instructions of two substantially disjoint instruction sets, a native instruction set providing access to substantially all of the resources of the computer, and a non-native instruction set providing access to a subset of the resources of the computer.

30. (original) The computer of claim 29, wherein the instruction pipeline and profile circuitry are further configured to effect recording of profile information describing an interval of the execution of an operating system coded in the non-native instruction set.

31. (new) The method of claim 1, wherein:

the recorded profile information describes at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution; and

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change taken together with a processor mode before the mode change instruction

32. (new) The method of claim 1, wherein:

the recorded profile information identifies each distinct physical page of instruction text executed during the execution interval.

33. (new) The method of claim 1, wherein:

the profile information records the address of the last byte of the multi-byte control transfer instruction.